NASA-CR-192744

CIRSSE
REPORT # 58

TECHNICAL REPORTS

NAGW-1333

(NASA-CR-192744) ON THE REVISED THEORY OF INTELLIGENT MACHINES (Rensselaer Polytechnic Inst.) 41 p

N93-71630

Unclas

Z9/63 0153778

153778

p. 41

CIRSSE

Center for Intelligent Robotic Systems for Space Exploration

Rensselaer Polytechnic Institute
Troy, New York 12180-3590

ON THE REVISED THEORY OF INTELLIGENT MACHINES

by

George N. Saridis

Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute
Troy, New York 12180-3590

June 1990

CIRSSE REPORT #58

ON THE REVISED THEORY OF INTELLIGENT MACHINES

George N. Saridis
Electrical, Computer and Systems Engineering Department
Rensselaer Polytechnic Institute
Troy, New York 12180-3590

April 13, 1990

ABSTRACT

After several years of experience the Theory of Intelligent Machines has been reformulated analytically to reflect the present state of the art. The functions of an Intelligent Machines are executed by Intelligent Controls. The Principle of Increasing Precision with Decreasing Intelligence is used to form a hierarchical structure of the control systems. Distributed Intelligence in compatible with such a structure when it is used for teams of intelligent machines or cooperating coordinators within the machine. The three levels of the Intelligent Control, e.g., the Organization, Coordination and Execution Levels are described as originally conceived. New designs as Neural-nets for the organization level and Petri- nets for the coordination level are included. Application to Intelligent Robots for space exploration has been focused in this work.

1 INTRODUCTION

Several researchers have made a considerable effort to develop viable theories for Intelligent Machines and create working models to implement such a theories (Albus 1985, Meystel 1986, Pao 1986, Saridis 1985, Zames 1979, etc.), in the past fifteen years. Such machines were designated to perform anthropomorphic tasks with minimum interaction with a human with potential applications on robotic systems designed to operate in remote, inaccessible, hazardous, unfamiliar or other environments as need appeared.

Since the task was enormous and the available technologies rather limited, the results of such an effort have been meager. The theoretic efforts that have come into the picture to reinforce the development of Intelligent Machines have taken two distinct directions: the logic-based approach (Nielsen-Genesereth 1988) and the analytic approach (Saridis 1988, Meystel 1986).

The results on the analytic approach, which concerns this particular paper, have been reported regularly by the author (Saridis 1977, Saridis 1979, Saridis 1983, Saridis 1985c, Saridis 1988) and have reached a level of maturity both theoretically and experimentally. A summary of the work produced is given in the next section.

In the past few years, new methodologies and new techniques like Neural-nets, Petri-nets, Boltzmann machines etc., have appeared in the literature and have provided new tools for the analytic formulation of the theory of Intelligent Machines. The further development and adaptation of such tools, along with the better understanding of the process led to modifications and refinements of the theory aimed to strengthen, simplify and integrate the proposed design of Intelligent Machines.

The refinements introduced herein are due to the better understanding of some of the basic concepts of the Intelligent Machines, e.g., the Principle of Increasing Precision with Decreasing Intelligence (IPDI), the ability to develop Boltzmann Machines and Petri-Nets as realizations of Inference Engines and Linguistic Decision Schemata, respectively, and the use of entropy measures for the evaluation of the performance at every level of the machine.

A review of the analytic formulation of the Intelligent Machines is given in the next section, followed by a set of pertinent definitions and a discussion on the principle of IPDI. A brief explanation of the development of the Boltzmann machine as an inference engine for the organization level is next. The following section presents details on the three levels of the Intelligent Machine with introduction of Neural-nets to the organization level and Petri-nets to the coordination level. The next section places the Intelligent Machine in a Distributed Intelligence environment, followed by applications, discussions and conclusions.

2 REVIEW OF THE ANALYTIC FORMULATION OF INTELLIGENT CONTROLS

Intelligent Machines require the use of "generalized" control efforts in order to perform intelligent functions such as simultaneous utilization of a memory, learning, or multilevel decision making in response to "fuzzy" or qualitative commands. Intelligent Controls have been developed by Saridis (1977, 1983) to implement such functions. For purposes of consistency of definition along the machine, "generalized" control is defined in a more liberal way as:

The process of making a system do what you want it to do.

Intelligent Control, utilize the results of cognitive systems research effectively with various mathematical programming control techniques.

Cognitive systems have been traditionally developed as part of the field of Artificial Intelligence to implement, on a computer, functions similar to ones encountered in human behavior (Albus 1975, Minsky 1972, Winston 1977, Nilsson 1969, Pao 1986). Such functions as speech recognition and analysis, image and scene analysis, data base organization and dissemination, learning and high-level decision making, have been based on methodologies emanating from a simple logic operation to advanced reasoning as in pattern recognition, linguistic and fuzzy set theory. The results have been well documented in the literature.

Various pattern recognition, linguistic or even heuristic methods have been used to analyze and classify speech, images or other information coming in through sensory devices as part of a cognitive system (Birk and Kelley 1981). Decision making and motion control were performed by a dedicated digital computer using either kinematic methods, like trajectory tracking, or dynamic methods based on compliance, dynamic programming or even approximately optimal control (Saridis and Lee 1979).

The theory of Intelligent Control systems, proposed by Saridis (1979) combines the powerful high-level decision making of the digital computer with advanced mathematical modeling and synthesis techniques of system theory with linguistic methods of dealing with imprecise or incomplete information. This produces a unified approach suitable for the engineering needs of the future. The theory may be thought of as the result of the intersection of the three major disciplines of Artificial Intelligence, Operations Research, and Control Theory as depicted in Figure 1. This research is aimed to establish Intelligent Controls as an engineering discipline, and it plays a central role in the design of Intelligent Autonomous Systems.

Intelligent Control can be considered as a fusion between the mathematical and linguistic methods and algorithms applied to systems and processes. They utilize the results of cognitive systems research effectively with various mathematical programming control techniques.

The control intelligence is hierarchically distributed according to the Principle of Precision with Decreasing Intelligence (IPDI), evident in all hierarchical management systems, and it is further discussed in a future section (Saridis 1988b). The resulting structure is composed of three basic levels of controls, each level of which may contain more than one layer of tree-structured functions (Saridis 1979) (See Figure 2):

1. The organization level.
2. The coordination level.
3. The execution level.

The functions involved in the upper levels of an intelligent machine are imitating functions of human behavior and may be treated as elements of knowledge-based systems. Actually, the activities of planning, decision making, learning, data storage and retrieval, task coordination, etc., may be thought of as knowledge handling and management. Therefore, the flow of knowledge in an intelligent machine may be considered as the key variable of such a system.

Knowledge flow in an intelligent machine represents respectively:

1. Data Handling and Management.
2. Planning and Decision performed by the central processing units.
3. Sensing and Data Acquisition obtained through peripheral devices.
4. Formal Languages which define the software.

Subjective probabilistic models are assigned to the individual functions. Their entropies may be evaluated for every task executed. This provides an analytical measure of the total activity.

Artificial Intelligence methods, using among other logic functions have been used to implement Intelligent Machines (Albus 1975, Meystel 1985, Nielsen Genesereth 1988). However, they lack that rigor and precision that analytic techniques provide. Nevertheless new methodologies have been adapted to analytic models to perform tasks at the various levels of an Intelligent Machine.

Moed and Saridis (1990), proposed a neural net approach to perform reasoning, planning and decision making in the organization level of an Intelligent Machine. A Boltzmann machine, suitable for the discrete binary state model of this particular level, is a natural device for organizing actions and rules necessary for the execution of a given command, regardless of the particular world model the machine is inhabiting.

Wang and Saridis (1988) proposed a Petri-net transducer to implement the Linguistic Decision Schemata (Saridis and Graham 1984), which serve as model coordinators and decision makers at the machine's coordination level. These devices set up the communication protocols, with the help of small real-time memories, and serve apply in real time the rules generated by the organization level to properly generate and sequence the subtasks in the particular environment of the machine, in order to execute the given original command.

Finally, Saridis (1988a) was able to reformulate the system control problem to use entropy as a control measure and therefore integrate all the hardware activities associated with the Intelligent Machine with the other levels regardless of the discipline they belong to. Thus, vision coordination, motion control, path planning, force sensing, etc., in a robot paradigm, may be integrated into the pertinent actions of the machine and evaluated by common entropy functions.

Since all levels of a hierarchical intelligent control can be measured by entropies and their rates, then the optimal operation of an "intelligent machine" can be obtained through the solution of mathematical programming problems.

Another development of this theory the structure of the "nested hierarchical" systems (Meystel, 1986). Even when the hierarchy is not tree-like, still using hierarchy is beneficial since the hierarchy of resolutions (errors per level) helps to increase the effectiveness of the system under limited computing power which is important to mobile systems.

The various aspects of the theory of hierarchically intelligent controls may be summarized as follows:

> The theory of intelligent machines may be postulated as the mathematical problem of finding the right sequence of decisions and controls for a system structured according to the principle of increasing precision with decreasing intelligence (constraint) such that it minimizes its total entropy.

The above analytic formulation of Intelligent Machines as a hierarchically intelligent control problem is based on the use of entropy as a meansure of performance at all the levels of the hierarchy. It has many advantages because of the tree-like structure of the decision making process, and brings together functions that belong to a variety of disciplines. The complete development of this theory and its integration with the other theoretical issues of the Intelligent Autonomous System is the main task of this paper.

3 SOME DEFINITIONS AND THE IPDI

3.1 Definitions

It remains to investigate the general concepts of Intelligent Control Systems which pertain to the fundamental functions Intelligent Machines. Such are the notions of Machine Knowledge, its Rate and Precision.

> Definition 1. Machine Knowledge is defined to be the structured information acquired and applied to remove ignorance or uncertainty about a specific task pertaining to the Intelligent Machine.

Knowledge is a cumulative quantity accrued by the machine and cannot be used as a variable to execute a task. Insteady, the Rate of Machine Knowledge is a suitable variable.

> Definition 2. Rate of Machine Knowledge is the flow of knowledge through an Intelligent Machine.

Intelligence is defined by the American Heritage Dictionary of the English Language (1969) as the capacity to acquire and apply knowledge.

In terms of Machine Intelligence, this definition may be modified to yield:

> Definition 3. Machine Intelligence (MI) is the set of actions or rules which operates on a data-based (DB) of events to produce flow of knowledge (R).

On the other hand, one may define Precision as follows:

> Definition 4. Imprecision is the uncertainty of execution of the various tasks of the Intelligent Machine.

and

> Definition 5. Precision is the complement of Imprecision, and represents the complexity of a process.

Analytically, the above relations may be summarized as follows:

Knowledge (K) representing a type of information may be represented as

$$K = -\alpha - ln p(K) \tag{1}$$

where $p(K)$ is the probability density of Knowledge.

From equation (1) the probability density function $p(K)$ satisfies the following expression in agreement with Jaynes' Principle of Maximum Entropy (1957):

$$p(K) = e^{-\alpha - K}; \quad \alpha = ln \int_X e^{-K} dx \tag{2}$$

The Rate of Knowledge R which is the main variable of an intelligent machine with discrete states is defined over a fixed interval of time T:

$$R = \frac{K}{T}$$

It was intuitively thought (Saridis 1983), that the Rate of Knowledge must satisfy the following relation which may be thought of expressing the principle of Increasing Precision with Decreasing Intelligence.

$$(MI) : (DB) \rightarrow (R) \tag{3}$$

A special case with obvious interpretation is, when R is fixed, machine intelligence is largest for a smaller data base, e.g., complexity of the process. This is in agreement with Vamos' theory of Metalanguages (1986).

It is interesting to notice the resemblance of this entropy formulation of the Intelligent Control Problem with the ε-entropy formulation of the metri theory of complexity originated by Kolomogorov (1956) and applied to system theory by Zames (1979). Both methods imply that an increase in Knowledge (feedback) reduces the amount of entropy (ε-entropy) which measures the uncertainty involved with the system.

An analytic formulation of the above principle derived from simple probabilistic relation among the Rate of Knowledge, Machine Intelligence and the Data Base of Knowledge, is presented in the next section. The entropies of the various functions come naturally into the picture as a measure of their activities.

3.2 The Analytic Formulation of IPDI

In order to formulate mathematically the concepts of knowledge-based systems, one must consider the state space of knowledge Ω, with states $s_i, i = 1, 2, ...n$. They represent the state of events at the nodes of a network defining the stages of a task to be executed.

Then knowledge between two states is considered as the association of the state s_i with another state s_j and is expressed as

$$K_{ij} = \frac{1}{2} w_{ij}' s_i s_j \tag{4}$$

where w_{ij} are state transition coefficients, which are zero in case of inactive transmission.

Knowledge at the state of s_i is the association of that stat with all the other active states s_j and is expressed as

$$K_i = \frac{1}{2} \sum_j w_{ij}' s_i s_j \tag{5}$$

Finally, the total knowledge of a system is considered as

$$K = \frac{1}{2}\sum_i \sum_j w'_{ij} s_i s_j \tag{6}$$

and has the form of energy of the underlying events. The rate (flow) of knowledge is the derivative of knowledge and for the discrete state space Ω_s is defined respectively

$$R_{ij} = \frac{K_{ij}}{T} \ , \ R_i = \frac{K_i}{T} \ , \ R = \frac{K}{T} \tag{7}$$

where T is a fixed time interval.

Since knowledge was defined as structured information, it can be expressed by a probabilistic relation similar to the one given by Shannon, and expressed for each level by equation (1):

$$lnp(K_i) = -\alpha - K_i \tag{8}$$

which yields a probability distribution satisfying Jaynes' Principle of Maximum Entropy (Jaynes 1957). For $E\{K\} = \text{Const.}$:

$$p(K_i) = e^{-a_1 - K_i}; \quad e^{a_1} = \sum_i e^{K_i}$$

The rate of knowledge is also related probabilistically by considering $K_i = R_i T$.

$$p(R_i) = p(R_i T) = e^{-\alpha_1 - TR_i} = e^{-\alpha_1 - \mu_1 R_i} \tag{9}$$

The principle of Increasing Precision with Decreasing Intelligence is expressed probabilistically by

$$PR(MI, DB) = PR(R) \tag{10}$$

where MI is the machine intelligence and DB is the data base associated with the task to be executed and represents the complexity of the task which is also proportional to the precision of execution. The following relation is obtained by conditioning and taking the natural logarithms:

$$lnp(MI/DB) + lnp(DB) = lnp(R) \tag{11}$$

Taking the expected value on both sides

$$H(MI/DB) + H(DB) = H(R) \tag{12}$$

where $H(x)$ is the entropy associated with x. For a constant rate of knowledge which is expected during the conception and execution of a task increase of the entropy of DB requires a decrease of the entropy of MI for the particular data base, which manifests the IPDI. IF MI is independent of DB then

$$H(MI) + H(DB) = H(R) \tag{13}$$

In the case that $p(MI)$ and $p(DB)$ satisfy Jaynes' Principle as $p(R)$ does, where

$$\begin{aligned} p(MI/DB) &= e^{\alpha_2 - \mu_2 MI_{DB}} \\ p(DB) &= e^{\alpha_3 - \mu_3 DB} \end{aligned} \tag{14}$$

where α_i and $\mu_i = 2,3$ are appropriate constants.
Then the entropies are rewritten as

$$-\alpha_2 - \mu_2 MI_{DB} - \alpha_3 - \mu_3 D\dot{B} = -\alpha_1 - \mu_1 R \tag{15}$$

and if

$$\alpha_1 = \alpha_2 + \alpha_3 \quad \gamma_2 = \frac{\mu_2}{\mu_1}, \ \gamma_3 = \frac{\mu_3}{\mu_1}$$

then

$$\gamma_2 MI_{DB} + \gamma_3 DB = R \tag{16}$$

which represents a specific but more explicit version of the Principle of Increasing Precision with Decreasing Intelligence. A detailed proof of the Principle is given in Saridis (1989).

This Principle is applicable both across one level of the Intelligent Hierarchy as well as throught the levels of the Hierarchy, in which case the flow R represents the throughput of the system in an information theoretic manner. The partition law of information rate applies naturally to such a system (Saridis 1985c).

The entropy of DB may be related to ε-entropy as follows: A system requiring certain (n) level of precision takes n-times the data base DB required for a simple precision. But

$$H(nDB) = Elnn + ElnDB \tag{17}$$

where $E\{lnn\}$ is the ε-entropy associated with the complexity of execution. As case study demonstrating the validity of the above is given in Saridis and Valavanis (1988).

4 THE ANALYTIC STRUCTURE OF THE INTELLIGENT MACHINE

In order to implement an Intelligent Machine on analytic foundations, the theory of Intelligent Control has been developed by Saridis (1979), and briefly discussed in Section 2. This theory assigns analytic models to the various levels of the machine and improve them through a generalized concept of selective feedback.

The Intelligent Control System is composed of three levels in decreasing order of intelligence and increasing order of precision as stipulated by the IPDI. However, with the better understanding of the basics, new methodologies are proposed to analytically implement the various functions, without significantly changing the models at each level.

The Organization Level is designed to organize a sequence of abstract actions or rules from a set of primitives stored in a long-term memory regardless of the present world model. In other words it serves as the generator of the rules of an Inference Engine by processing (intelligence) high level of information, for reasoning, planning and decision making. This can be accomplished by a two level Neural-net, analytically derived as a Boltzmann machine by Saridis and Moed (1988 and 1990).

The Coordination Level is an intermediate structure serving as an interface between the organization and execution levels. It deals with real- time information of the world by generating a proper sequence of subtasks pertinent to the execution of the original command.

It involves coordination of decision making and learning on a short term memory, e.g., a buffer. It utilizes Linguistic Decision Schemata with learning capabilities defined in Saridis and Graham (1984), assigned subjective probabilities for each action. The respective entropies may be obtained directly from these subjective probabilities. Petri Net Transducers have been investigated by Wang and Saridis (1988), to implement such decision schemata. In addition, Petri-nets provide the necessary protocols to communicate among the various coordinators, in order to integrate the activities of the Machine. Complexity functions may be used for real-time evaluation.

The Execution Level performs the appropriate control functions on the processes involved. Their performance measure can also be expressed as an entropy, thus unifying the functions of an Intelligent Machine.

Optimal control theory utilizes a non-negative functional of the states of a system in the state space, and a specific control from the set of all admissible controls, to define the performance measure for some initial conditions, representing a generalized energy function. Minimization of the energy functional yields the desired control law for the system.

The Principle of IPDI is applicable at every level of the Machine, reaffirming its universal validity. However, the coordination may serve as a salient example of its application where the intelligence provided by the organization level as a set of rules is applied to the database provided by the execution level to produce flow of knowledge.

A more detailed description of the analytic functions of each level is given in the sequel.

4.1 The Neural-Net Based Organization Level

The function of the organizer, the highest level of the hierarchy of Intelligent Controls, is based on several AI (knowledge based) concepts forming the foundations of Machine Intelligence. These concepts translated into probabilistic models form the functions of representation and reasoning, planning, decision making, long-term memory exchange and learning through feedback to set up a task in response to some outside command (Figure 3). The probabilistic model generated provides the mechanism to select the appropriate task for the appropriate command. The principle followed here is that instead of task decomposition a collection of tasks is generated from a list of primitives stored in the memory and matched against the input command applied.

To specify analytically the model of the organizer, it is essential to derive the domain of the operation of the machine for a particular class of problems (Saridis and Valavanis 1988). Assuming that the environment is known, one may define the following functions on the organization level:

a) <u>Machine Representation and Abstract Reasoning, (RR)</u> is the association of the compiled command to a number of activities and/or rules. A probability function is assigned to each activity and/or rule and the Entropy associated with it is calculated. When rules are included one has active reasoning (inference engine).

   In order to generate the required analytic model of this function the following sets are defined:

The set of commands $C = \{c_1, c_2, ..., c_m\}$ in natural language, is received by the machine as inputs. Each command is compiled to yield an equivalent machine code explained in the next section.

The task command of the machine contains a number n of independent events.

The events $E = \{e_1, e_2, ..., e_n\}$ are individual primitive objects or actions e_i stored in the long-term memory and repesenting primitive tasks to be executed. The task domain indicates the capabilities of the machine. Events represent the nodes of the Neural-net.

Activities A, are groups of events concatenated to define a complex task; e.g., $A_{234} = \{e_2, e_3, e_4\}$. If the events are ordered then we have an ordered activity.

A set of random variables $X = \{x_1, \ldots x_n\}$ representing the state of events is associated with each individual event e_i. If the random variable x_i is binary (either 0 or 1), it indicates whether an event e_i is inactive or active, in a particular activity and for a particular command. If the random variables x_i are continuous (or discrete but not binary) over [0,1], they reflect a membership function in a fuzzy decision making problem. At this point, the x_i's are considered to be binary.

A set of probabilities P associated with the random variables X is defined as follows.

$$P = \{p_i = Prob[x_i = 1]\}$$

The probabilities P are known at the beginning of the representation stage. In order to reduce the problem of dimensionality a subset of events is defined for the given command c_k.

$$S_k = \{e_i; p_i > a\} \subset E$$

b) Machine Planning, (P), is ordering of the activities.

The ordering is obtained by properly concatenating the appropriate abstract primitive events $e_i \subset S_k$ for the particular command c_k, in order to form the right abstract activities (sentences or text).

The ordering is generated by a Boltzmann machine which measures the average flow of knowledge from node j to node i on the Neural-net by

$$R_{ij} = \frac{1}{2}E\{w_{ij}x_ix_j\} = \frac{1}{2}w_{ij}p_ip_j \geq 0 \tag{18}$$

The probability due to the uncertainty of knowledge flow into node i, is calculated as in (9):

$$p(R_i) = \bar{\epsilon}^{\alpha_i - \frac{1}{2}\sum_j w_{ij}p_ip_j} \tag{19}$$

where

$w_{ij} \geq 0$ is the interconnection weight between nodes i and j

$w_{ij} = 0$

$\alpha_i > 0$ is a probability normalizing factor.

The average Flow of Knowledge R_i into node i, is:

$$R_i = \alpha_i + \frac{1}{2}E\{\sum_j w_{ij}x_ix_j\} = \alpha_i + \frac{1}{2}\sum_j w_{ij}p_ip_j$$

with probability $P(R_i)$, (Jaynes' Principle):

$$P(R_i) = exp[-\alpha_i - \frac{1}{2}\sum_j w_{ij}p_ip_j].$$

The Entropy of Knowledge Flow in the machine is

$$H(R) = -\sum_i P(R_i)\text{in}[P(R_i)] = \sum_i(\alpha_i + \frac{1}{2}\sum_j w_{ij}p_ip_j)exp[-\alpha_i - 1/2\sum_j w_{ij}p_ip_j.] \tag{20}$$

The normalizing factor α_i is such that $1/2^n \leq P(R_i) \leq 1$.

The entropy is maximum when the associated probabilities are equal, $p(R_i) = \frac{1}{2^n}$ with n the number of nodes of the network. By bounding $p(Ri)$ from below by $\frac{1}{2^n}$ one may obtain a unique minimization of the entropy corresponding to the most like sequence of events to be selected.

Unlike the regular Boltzmann machines, this formulation does not remove α_i when $p_i = 0$. Instead, the machine operates from a base entropy level $\propto_i \epsilon^{-\alpha_i}$ defined as the Threshold Node Entropy which it tries to reduce (Saridis and Moed, 1988).

c. Machine Decision Making, (DM) is the function of selecting the sequence with the largest probability of success.

This is accomplished through a search to connect a node ahead that will minimize the Entropy of Knowledge Flow at that node:

$$H(R_i) = (\alpha_i + \frac{1}{2}\sum_j w_{ij}p_ip_j)\ exp[-\alpha_i - \frac{1}{2}\sum_j w_{ij}p_ip_j]$$

A modified genetic algorithm, involving a global random search, has been proposed by Moed and Saridis (1990) as a means of generating the best sequence of events that minimized the uncertainty of connections of the network expressed by the entropy (20).

This algorthim, proven to converge globally compared favorably with other algorithms like the Simulated Annealing and the Random Search. The DM process is illustrated in Figure 4.

d. Machine Learning, (ML) (Feedback). Machine Learning is obtained by feedback devices that upgrade the probabilities p_i and the weights w_{ij} by evaluating the performance of the lower levels after a successful iteration.

For y_k representing either p_i or w_{ij}, corresponding to the command c_k, the upgrading algorithms are:

$$\begin{aligned} y_k(t_k+1) &= y_k(t_k) + \beta_k(t_k+1)[\xi(t_k+1) - y_k(t_k)] \\ J_k(t_k+1) &= J_k(t_k) + \gamma(t_k+1)[V_{obs}^k(t_k+1) - J_k(t_k)] \end{aligned} \qquad (21)$$

where J_k is the performance estimate, V_{obs}^k its observed value and

$$\begin{aligned} p_i &: \ \xi_k(t_k+1) = x(t_k) \\ w_{ij} &: \ \xi_k(t_k+1) = \begin{cases} 1 & \text{if } J = \min_e J_e \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{22}$$

e. Memory Exchange (ME), is the retrieval and storage of information from the long-term memory, based on selected feedback data from the lower levels after the completion of the complex task.

The above functions may be implemented by a two level Neural-net, of which the nodes of the upper level represent the primitive objects e_{oi} and the lower level of primitive actions relating the objects e_{ai} of a certain task. The purpose of the organizer may be realized by a search in the Neural-net to connect objects and actions in the most likely sequence for an executable task.

4.2 The Coordination Level and Petri Net Transducers

The coordination level is an intermediate structure serving as an interface between the organization and the execution level. It is essential for dispatching and communicating organizational information to the execution level. Its objective is the actual formulation of the control problem associated with the most probable complete and compatible plan generated by the organization level and based on real-time acquired information about the world model.

The functions of the coordination level are summarized as follows:

a. Dispatching of tasks requested by the organization level.

b. Identification and of the current environment.

c. Real-time decision-making.

d. Data transfer and communication.

e. Use of formal languages.

f. Learning (Feedback)

g. Interfacing.

The coordination level is composed of a dispatcher and a number of specialized coordinators (Figure 5). Specific hardware (execution devices) from the execution level is associated with each coordinator. These devices execute well defined tasks when a command is issued to them by their corresponding coordinator (Saridis and Valavanis 1988). The dispatcher serves as both the communicator of information from the organization level to the coordinators and on-line exchange of data among the coordinators. A Petri-net formulation of these activities has been recently proposed by Wang and Saridis (1988).

Petri-nets have been proposed as devices to communicate and control complex heterogenous processes. These nets provide a communication protocol among stations of the process as well as the control sequence for each one of them (Peterson 1977).

Abstract task plans, suitable for many environments are generated at the organization level by a grammar:

$$G = (N, \textstyle\sum_o, P, S)$$

where

$N = \{S, M, Q, H\}$ = Non-terminal symbols

$\sum_o = \{A_1, A_2 \ldots A_n\}$ = Terminal Symbols (activities)

P = Production rules

Petri Net Transducers (PNT) proposed by Wang and Saridis (1988) are Petri-net realizations of the Linguistic Decision Schemata introduced by Saridis and Graham (1984) as linguistic decision making and sequencing devices. They are defined as 6-tuples.

$$M = (N, \textstyle\sum, \Delta, G, \mu, F)$$

where

$N = (P, T, I, O)$ = A Petri-net with initial marking μ

$\sum$ = a finite input alphabet

Δ = a finite output alphabet

σ = a translation mapping from T x $(\sum \cup \{\lambda\})$ to finite sets of Δ^* and $F \subset R(\mu)$ a set of final markings.

A Petri Net Transducer is depicted in Figure 6. Its input and output languages are Petri Net Languages (PNL). In addition to its on-line decision making capability PNT's have the potential of generating communication protocols, learning by feedback, ideal for the communication and control of coordinators and their dispatcher in real time. Their architecture is given in Figure 7, and may follow a scenario suitable for the implementation of an autonomous intelligent robot.

Figure 8 depicts the Petri-net Structure of a typical coordination structure (CS) of an intelligent robot. This structure is a 7-tuple:

$$CS = (D, C, F, R_D, S_D, R_C, S_C)$$

where

$D = (N_d, \sum_o, \Delta_o, G_d, \mu_d, F_d)$ = The PNT dispatcher

$C = \{C_1, \ldots C_n\}$ = set of coordinators

$C_i = (N_c^i, \sum_c^i, \Delta_c^i, G_c^i, F_c^i)$ = the ith PNT coordinator

$F = \bigcup_{i=1}^n \{f_I^i, f_{SI}^i, f_o^i, f_{SO}^i\}$ = set of connection points

R_D, R_C = Receiving maps for the dispatcher and coordinators

S_D, S_C = Sending maps for the dispatcher and coordinators

Decision making in the coordination structure is accomplished by Task Scheduling and Task Translation, e.g., for a given task find σ an enabled t such that $\sigma(t, a)$, is defined and then select the right translation string from $\sigma(t, a)$ for the transition t.

The sequence of events transmitted from the organization level is received by the dispatcher which requests a world model with coordinates from a vision coordinator.

The vision coordinator generates appropriate database and upon the dispatcher's command communicates it to the planning coordinator which set a path for the arm manipulator. A new command from the dispatcher sends path information to the motion controller in terms of end points, constraint surface and performance criteria. It also initializes the force sensor and proximity sensor control for grasp activities. The vision coordinator is then switched to a monitoring mode for navigation control, and so on.

The PNT can be evaluated in real-time by testing the computational complexity of their operation which may be expressed uniformly in terms of entropy. Feedback information is communicated to the coordination level from the execution level during the execution of the applied command. Each coordinator, when accessed, issues a number of commands to its associated execution devices (at the execution level). Upon completion of the issued commands feedback information is received by the coordinator and is stored in the short-term memory of the coordination level. This information is stored in the short-term memory of the coordination level. This information is used by other coordinators if necessary, and also to calculate the individual, accrued and overall accrued costs related to the coordination level. Therefore, the feedback information from the execution to the coordination level will be called <u>on-line</u>, <u>real-time</u> feedback information.

The performance estimate and the associated subjective probabilities are updated after the k_{ij}-th execution of a task $[(u_t, x_t)_i, S_j]$ and the measurement of the estimate of the observed cost J_{ij}:

$$J_{ij}(k_{ij}+1) = J_{ij}(k_{ij}) + \beta(k_{ij}+1)[J_{obs}(K_{ij}+1) - J_{ij}(k_{ij})] \tag{23}$$

$$p_{ij}(k_{ij}+1) = p_{ij}(k_{ij}) + \gamma(k_{ij}+1)[\xi_{ij}(k_{ij}+1) - p_{ij}(k_{ij})]$$

where

$$\xi_{ij} = \begin{cases} 1 & J_{ij} = \min \\ 0 & \text{elsewhere} \end{cases}$$

and β and γ are harmonic sequences. Convergence of this algorithm is proven in (Saridis and Graham 1984).

The learning process is measured by the entropy associated to the subjective probabilities. If

$$H(M) = H(E) + H(T/E) \tag{24}$$

where $H(E)$ is the environmental uncertainty and $H(T/E)$ is the pure translation uncertainty. Only the latter may be reduced by learning.

4.3 The Execution Level With Entropy Measures

The cost of control at the hardware level can be expressed as an entropy which measures the uncertainty of selecting an appropriate control to execute a task. By selecting an optimal control, one minimizes the entropy, e.g., the uncertainty of execution. The entropy may be viewed in the respect as an energy in the original sense of Boltzmann, as in Saridis (1988).

Optimal control theory utilizes a non-negative functional of the state of the system $x(t)\epsilon\Omega_x$ the state space, and a specific control $u(x,t)\epsilon\Omega_u xT; \Omega_u \subset \Omega_x$ the set of all admissible feedback controls, to define the performance measure for some initial conditions $x_0(t_0)$, representing a generalized energy function, of the form

$$V(x_0, t_0) = E\{\int_{t_0}^{t_f} L(x, t, u(x,t))dt\} \tag{25}$$

where $L(x,t,u(x,t)) > 0$, subject to differential constraints dictated by the underlying process

$$\frac{dx}{dt} = f(x, u(x,t), w, t)x(t_o) = x$$

$$z = g(x, v, t)x(t_f)\epsilon M_f \tag{26}$$

where $x_o, w(t), v(t)$ are random variables with associated probability densities $p(x_o)$, $p(w(t))$, $p(v(t))$ and M_f a maninfold in Ω_x. The trajectories of the system (26) are defined for a fixed but arbitrarily selected control $u(x,t)$ from the set of admissible feedback controls Ω_u.

In order to express the control problem in terms of an entropy function, one may assume that the performance measure $V(x_0, t_0, u(x,t))$ is distributed in Ω_u according to the probability density $p(u(x,t))$ of the controls $u(x,t)\epsilon\Omega_u$. The differential entropy $H(u)$ corresponding to the density is defined as

$$H(u) = -\int_{\Omega_u} p(u(x,t)) ln p(u(x,t))dx$$

and represents the uncertainty of selecting a control $u(x,t)$ from all possible admissible feedback controls Ω_u. The optimal performance should correspond to the maximum value of the associated density $p(u(x,t))$. Equivalently, the optimal control $u^*(x,t)$ should minimize the entropy function $H(u)$.

This is satisfied if the density function is selected to satify Jaynes' Principle of Maximum Entropy (1956), e.g.,

$$p(u(x,t)) = \exp\{-\lambda - \mu V(x_o, t_o, u(x,t))\} \tag{27}$$

where λ and μ are normalizing constants.

It was shown by Saridis (1985b) that the expression $H(u)$ representing the entropy for a particular control action $u(x,t)$ is given by

$$\begin{aligned} H(u) &= \int_{\Omega_u} p(x, u(x,t)) V(x_o, t_o, u(x,t))dx \\ &= \lambda + \mu V(x_0, t_0, u(x,t))\} \end{aligned} \tag{28}$$

This implies that the average performance measure of a feedback control problem corresponding to a specifically selected control, is an entropy function. The optimal control $u^*(x,t)$ that minimizes $V(x,t,u(x,t))$, maximizes $p(x,u(x,t))$, and consequently minimizes the entropy $H(u)$.

$$\begin{aligned} u^*(x,t) \;&:\; E\{V(x_o, t_o, u^*(x,t))\} \\ &= \min_u \int_{\Omega_u} V(x_o, t_o, u(x,t)) p(u(x,t))dx \end{aligned} \tag{29}$$

This statement is the generalization of a theorem proven in (Saridis 1988) and establishes equivalent measures between information theoretic and optimal control problem and provides the information and feedback control theories with a common measure of performance.

This optimal control theory designed mainly for motion control, can be implemented for vision control, path planning and other sensory system pertinent to an Intelligent Machine by slightly modifying the system equations and cost functions. After all one is dealing with real-time dynamic systems which may be modeled by a dynamic set of equations.

5 DISTRIBUTED MACHINE INTELLIGENT SYSTEMS

In the real world, distributed systems and hierarchical system co-exist in harmony. The human organism is a typical example of this statement.

Distributed Artificial Intelligence (DAI) is a discipline concerned with treating problems that require multiple solvers in parallel by invoking artificial intelligence techniques (Decker 1987). When utilized to control intelligent machines working in parallel, it can be interpreted as Distributed Machine Intelligence (DMI) where the intelligence processing is referred to the autonomous abilities of the machines involved as with simple hierarchically intelligent control case (Saridis 1986): This corresponds more to the distributed problem solving process and may be thought of as composed of two components:

Distributed Machine Intelligence:

1. Control,
2. Communications.

Distributed Control can be performed in two different ways:

1. Control by a meta level,
2. Control by majority vote.

The first method is an extension of the hierarchical approach where the coordination, decision making and subtask assignment is deferred to a higher level of intelligence imbedded in the dispatcher of the intelligent machines (see Figure 5). The cooperative activities should be planned, scheduled and sequenced in this device and communicated continuously to the machines. Feedback from the environment should be communicated continosusly for the evaluation of the team work performed.

The second method deals with cooperative approach of machines operating in the same environment and performing tasks that require scheduling and task assignment. Majority vote may provide the proper planning and sequencing of the various tasks to be performed in unison by all the intelligent machines involved. The majority vote could be taken in a poll place equally accessed by all the machine and communicated back to them in the appropriate sequence.

The communication problem plays a paramount role in distributed machine intelligence. It may be performed by a large communication network in the case of wide spacially distributed machines or by a computer bus when dealing with a tightly built system of devices. The main design considerations of a communication system are:

1. The system configuration,
2. The protocol, and
3. The treatment of uncertainty of information.

The first item deals with the selection of the proper structure of the network. Two types suitable for the appropriate control categories are:

1. Star Connection.
2. Ring Connection.

The second item is essential for the most efficient operation of the system and the optimization of the information exchange among the intelligent machines. The computer literature contains many sources of information about protocols as in Lampson, Paul and Siegert (1981).

The third item deals with ability of the communications system to deal with uncertain and incomplete information. The problem of reliability for accurate and precise transmission and reception of information is essential. The classical Shannon's information theory methods are applicable here (Shannon and Weaver, 1963).

Finally, as mentioned earlier, distributed machine intelligence may be applied to coordinate a number of cooperating intelligent machines or to organize a number of coordinators within the same machine. In both cases, such a structure can work in harmony with the hierarchically intelligent control structure of Saridis (1983). The reason is that the hierarchical stratification refers to the intelligence of the machine and the IPDI needs only to be generalized from a vertical to a horizontal deployment. In other words, the IPDI should be assigned to all directions of flow of knowledge to represent all the trade-offs between intelligence and complexity.

6 APPLICATION TO ROBOTIC SYSTEMS

The theory of Intelligent Controls has direct application to the design of Intelligent Robots. The IPDI provides a means of structuring hierarchically the levels of the machine. Since for a passive task the flow of knowledge through the machine must be constant, it assigns the highest level with the highest machine intelligence and smallest complexity (size of data base), and the lowest level with the lowest machine intelligence and largest complexity. Such a structure agrees with the concept of most organizational structures encountered in human societies. Application to machine structures is straight forward.

Even at the present time there is a large variety of applications for intelligent machines. Automated material handling and assembly in an automated factory, automation inspection, sentries in a nuclear containment are some of the areas where intelligent machines have and will find a great use. However, the most important application for the author's group is the application of Intelligent Machinesto unmanned space exploration where, because of the distance involved, autonomous anthropomorphis tasks must be executed and only general commands and reports of executions may be communicated.

Such tasks are suitable for intelligent robots capable of executing anthropomorphic tasks in unstructured uncertain environments. They are structured uncertain environment. They are structured usually in a human-like shape and are equipped with vision and other tactile sensors to sense the environment, two areas to execute tasks and locomotion for appropriate mobility in the unstructured environment. The controls of such a machine are performed according to the theory of Intelligent Machines previously discussed (Saridis and Stephanou 1977), (Saridis 1983, 1985a, 1985b, 1988a), (Meystel 1985, 1986). The three levels of controls, obeying the Principle of Increasing Precision with Decreasing Intelligence, are presently tested on a testbed composed of two PUMA 600 robot arms with stereo vision and force sensing, with the structure of Figure 9.

Recent research has been focused in the application of the Theory of Intelligent Machines to design robots for autonomous manipulation and locomotion in space. Satellite maintenance, construction of the space station and autonomous planet exploration vehicles are typical examples. A testbed for earth simulation of such activities in space has been built in the Center for Intelligent Robotics for Space Exploration at Rensselaer and graphically depicted in Figure 10.

7 CONCLUSIONS

A revision of the analytic formulation of Intelligent Machines developed by the author and his colleagues, has been proposed in this paper. The realization of the Machine is obtained through the application of Hierarchically Intelligent Control based on a better understanding of the Principle of Increasing Precision (complexity) with Decreasing Intelligence, which utilizes a three level structure. The upper level is implemented through a Boltzmann machine capable of task planning at an abstract level. The coordination level composed of a dispatcher and several coordinators are implemented by Petri Net Transducers, as realization of Linguistic Decision Schemata. Finally, the execution level may be modeled by a set dynamic system with entropy as a cost function, unifying the evaluation various processes.

Optimality is still searched at the individual levels. Total optimization, a mathematical programming problem is still to be investigated.

The application of interest to the author is presently Intelligent Machines for unmanned space exploration.

ACKNOWLEDGEMENTS

This work was supported by NASA Grant NAGW-1333.

REFERENCES

Albus, J.S. (1975), "A New Approach to Manipulation Control: The Cerebellar Model Articulation Controller", *Transactions of ASME, J. Dynamics Systems, Measurement and Control, 97*, pp. 220-227.

Athans, M. (1987), "Command and Control (c2) Theory: A Challenge to Control Science", *IEEE Trans. on AC, 32*, No. 4, pp. 286-293, April.

Birk, J.R. and Kelley, R.B. (1981), "An Overview of the Basic Research Needed to Advance the State of Knowledge in Robotics", *IEEE Trans. on SMC*, SMC-11, No. 8, pp. 575-579.

Decker, K.S. (1987), "Distributed Problem Solving Techniques, A Survey", *IEEE Trans. on Syst., Man and Cybernetics*, SMC-17, No. 5, 729- 40, September-October.

Fu, K.S. (1971), "Learning Control Systems and Intelligent Control Systems: An Intersection of Artificial Intelligence and Automatic Control", *IEEE Trans on Automatic Control*, Vol. AC-16, No. 1, pp. 70-72.

Hayes-Roth, F., et al. (1983), *Building Expert Systems*, Addison-Wesley, New York.

Hinton, G.E. and Sejnowski, T.J. (1986), "Learning and Relearning in Boltzmann Machines", pp. 282-317, in *Parallel Distributed Processing*, ed D.E. Rumelhart and J.L. McClellan, MIT Press.

Jaynes, E.T. (1957), "Information Theory and Statistical Mechanics", *Physical Review*, pp. 106, 4.

Kolmogorov, A.N. (1956), "On Some Asymptotic Characteristics of Completely Bounded Metric Systems", Dokl Akad Nank, SSSR, Vol. 108, No. 3, pp. 385-389.

Lampson, B.W., Paul, M., and Siegert, H., Eds. (1981), *Distributed Systems Architecture and Implementation*, Springer Verlag, New York.

Meystel, A. (1986), "Cognitive Controller for Autonomous Systems", *IEEE Workshop on Intelligent Control 1985*, p. 222, RPI, Troy, New York.

Minsky, M.L. (1972), *Artificial Intelligence*, McGraw-Hill, New York.

Moed, M.C. and Saridis, G.N. (1990), "A Boltzmann Machine for the Organization of Intelligent Machines", *IEEE Transactions on Systems Man and Cybernetics*, Vol. 20, No. 5, Sept.

Nilsson, N.J. (1969), "A Mobile Automation: An Application of Artificial Intelligence Techniques", *Proc. Int. Joint Conf. on AI*, Washington, D.C.

Pao, Y.-H (1986), "Some Views on Analytic and Artificial Intelligence Approaches", *IEEE Workshop on Intelligent Control*, p. 29, RPI, Troy, New York.

Peterson, J.L. (1977), "Petri-Nets", *Computing Survey, 9*, No. 3, pp. 223-252, September.

Sage (1987), "Information Systems Engineering for Distributed Decision Making". *IEEE Transactions on Systems Man and Cybernetics*, SMC-17, No. 6, pp. 920-36, Nov./Dec.

Saridis, G.N. (1977), *Self-Organizing Controls of Stochastic Systems*, Marcel Dekker, New York, New York.

Saridis, G.N. (1979), "Toward the Realization of Intelligent Controls", *IEEE Proceedings*, Vol. 67, No. 8.

Saridis, G. N. (1983), "Intelligent Robotic Control", *IEEE Trans. on AC-29*, 4.

Saridis, G.N. (1985a), "Intelligent Control-Operating Systems in Uncertain Environments", Chapter 7, in *Uncertainty and Control*, J. Ackermann Editor, Springer-Verlag, Berlin, pp. 215-233.

Saridis, G.N. (1985b), "Control Performance as an Entropy", *Control Theory and Advanced Technology*, 1, 2.

Saridis, G.N. (1985c), "Foundations of Intelligent Controls", *Proceedings of IEEE Workshop on Intelligent Controls*, p. 23, RPI, Troy, New York.

Saridis, G.N. (1988), "Entropy Formulation for Optimal and Adaptive Control", *IEEE Transactions on AC*, Vol. 33, No. 8, pp. 713-721.

Saridis, G.N. (1989), "Analytic Formulation of the IPDI for Intelligent Machines", *Automatic the IFAC Journal*, Vol. 25, No. 3, pp. 461-467.

Saridis, G.N. and Graham, J.H. (1984), "Linguistic Decision Schemata for Intelligent Robots", *Automatica*, Vol. 20, No. 1, 121-126.

Saridis, G.N. and Lee, C.S.G. (1979), "Approximation of Optimal Control for Trainable Manipulators", *IEEE Trans. on SMC*, Vol. SMC-8, No. 3.

Saridis, G.N. and Moed, M.C. (1988), "Analytic Formulation of Intelligent Machines as Neural Nets", Symposium on Intelligent Control, Washington, D.C., August.

Saridis, G.N. and Stephanou, H.E. (1977), "A Hierarchical Approach to the Control of a Prosthetic Arm", *IEEE Trans. on SMC*, Vol. SMC-7, No. 6, pp. 407-420.

Saridis, G.N. and Valavanis, K.P. (1988), "Analytical Design of Intelligent Machines", *Automatica the IFAC Journal*, Vol. 24, No. 2, pp. 123-133.

Sannon, C. and Weaver, W. (1963), *The Mathematical Theory of Communications*, Illini Books.

Vamos, T. (1986) "Metalanguages - Conceptual Models, Bridge Between Machine and Human Intelligence", Working Paper E/37, Hungarian Academy of Science.

Wang, F., Kyriakopoulos, K., Tsolkas T., Saridis, G.N., (1990) "A Petri-Net Coordination Model of Intelligent Mobile robots" CIRSSE Technical Report #50, Jan. 1990.

Wang, F. and Saridis, G.N. (1988), "A Model for Coordination of Intelligent Machines Using Petri Nets", Symposium on Intelligent Control, Washington, D.C., August.

Wang, F., Saridis, G.N. (1990) "A Coördination Theory for Intelligent Machines" Automatica the IFAC Journal, Vol. 35, No. 5, Sept.

Winston, P. (1977), *Artificial Intelligence*, Addison-Wesley, New York.

Zames, G. (1979), "On the Metric Complexity of Casual Linear Systems, ε-entropy and ε-dimension for Continous Time", *IEEE Trans. Automatic Control*, Vol. AC-124, pp. 220-230, April.

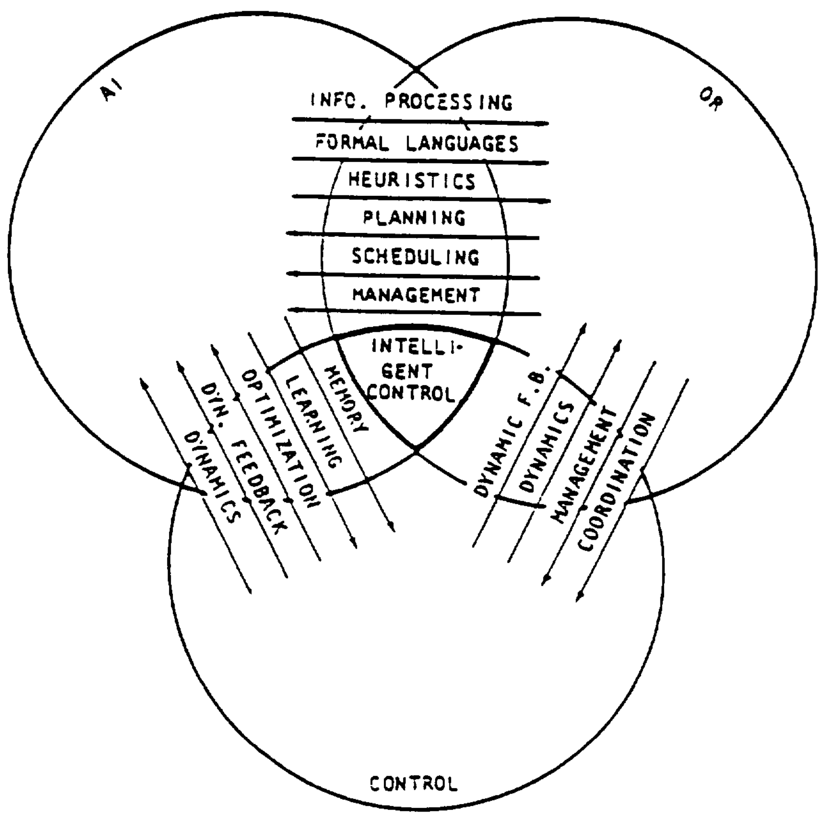


Fig. 1. Intersection of Artificial Intelligence Operations Research, and Control Theory and the Resulting Intelligent Control.

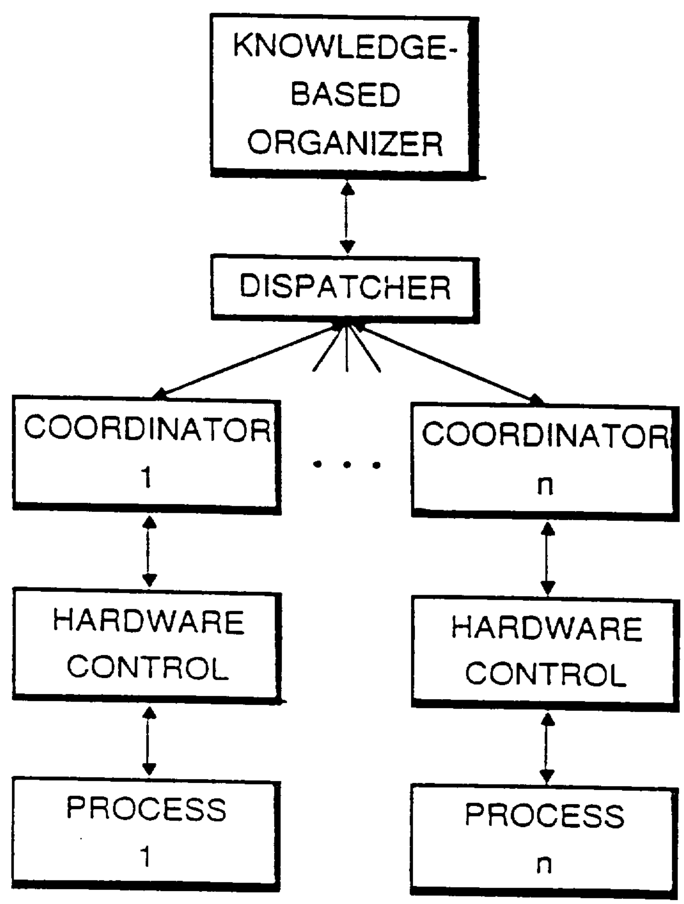


Fig. 2. Hierarchical Intelligent Control System.

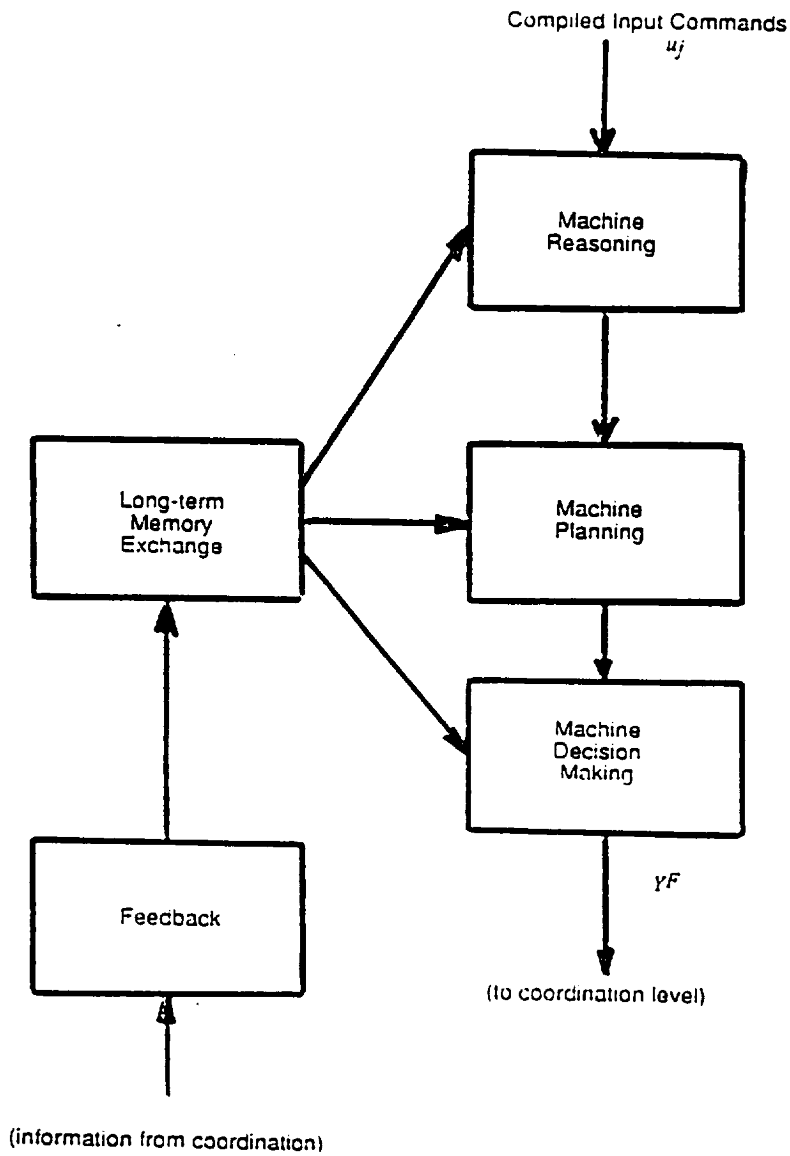


Figure 3. Block Diagram of the Organization Level

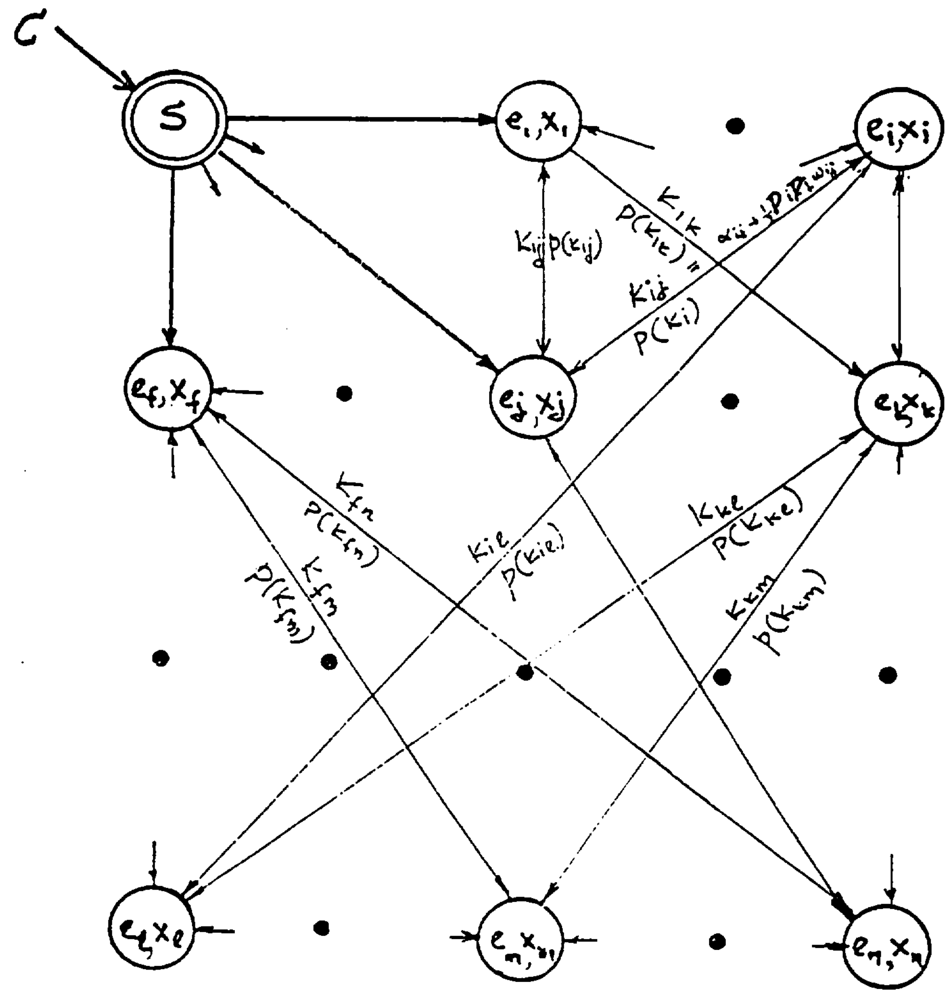


Fig. 4. The Boltzmann Machine for the Organization Level.

e_i = primitive event

x_i = state of event e_i, $\in \{1, 0\}$: with prob p_i

K_i = energy at node i. $= \alpha_i + \frac{1}{2} \sum_j p_i \, p_j \, w_{ij}$

w_{ij} = learned weights

$p(K_i)$ = probability of connection i-j.

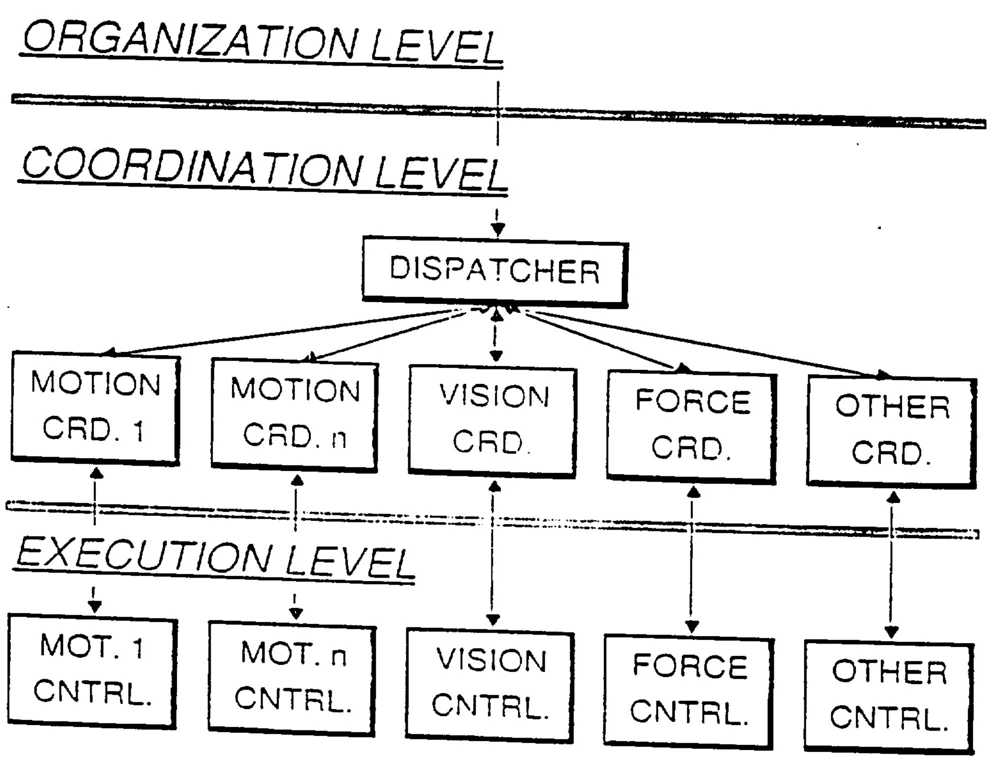


Fig. 5. Hierarchical Intelligent Manipulator Control System.

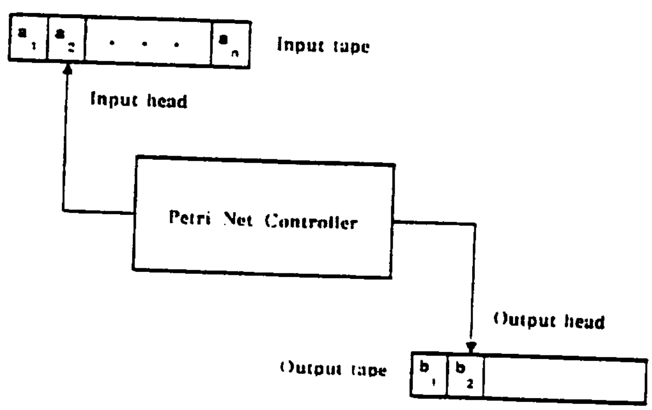


Figure 6: The Petri Net Transducer (PNT)

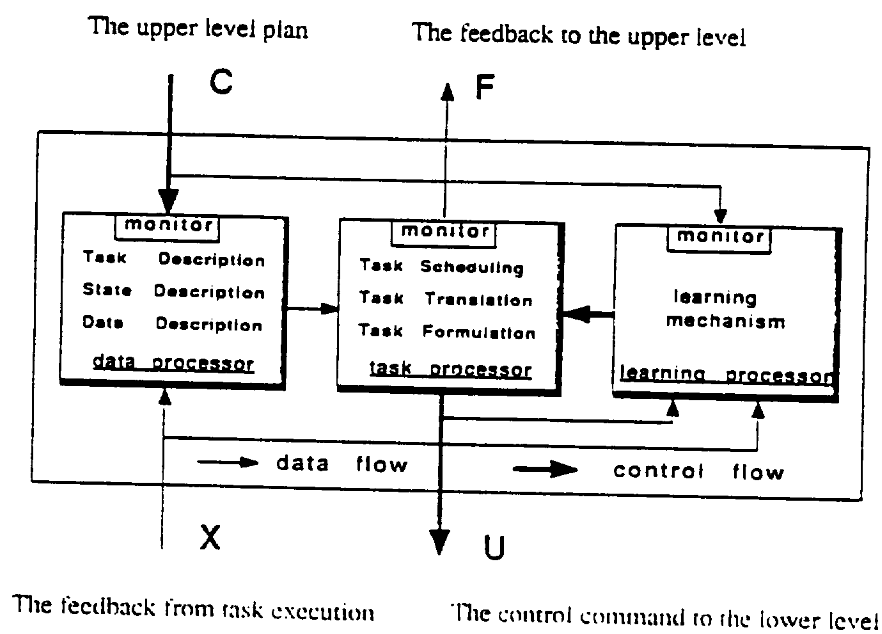


Figure 7: A Uniform Architecture for the Dispatcher and the Coordinators

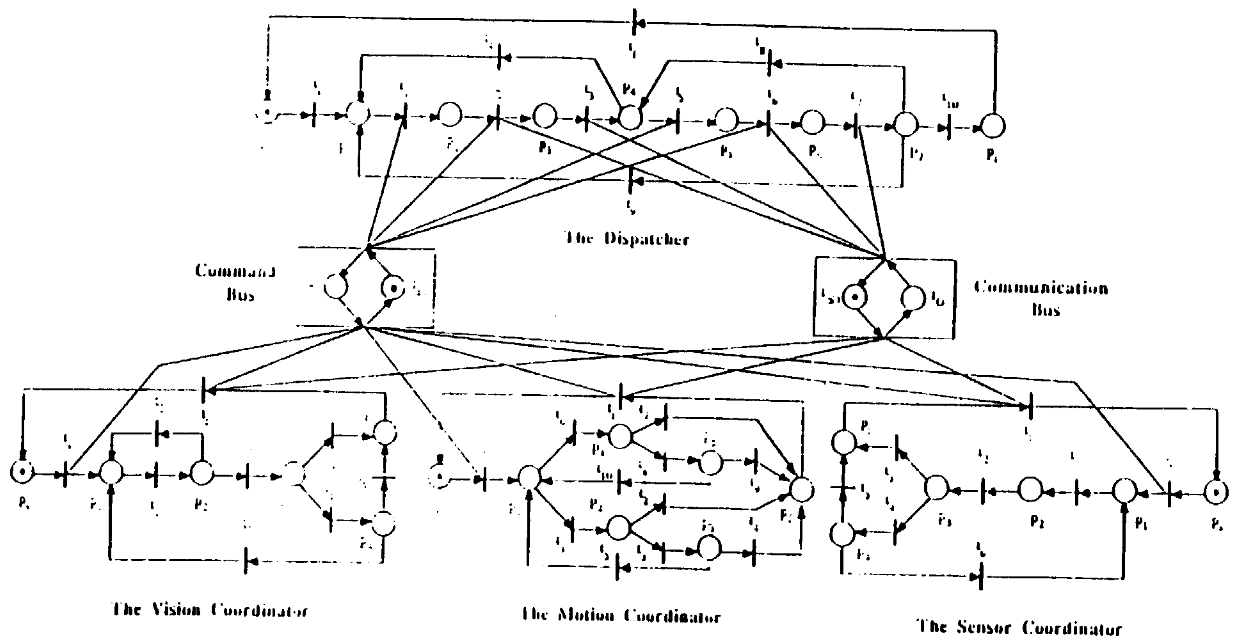


Figure 5 : The Coordination Structure of An Intelligent Manipulator System

ORIGINAL PAGE IS
OF POOR QUALITY

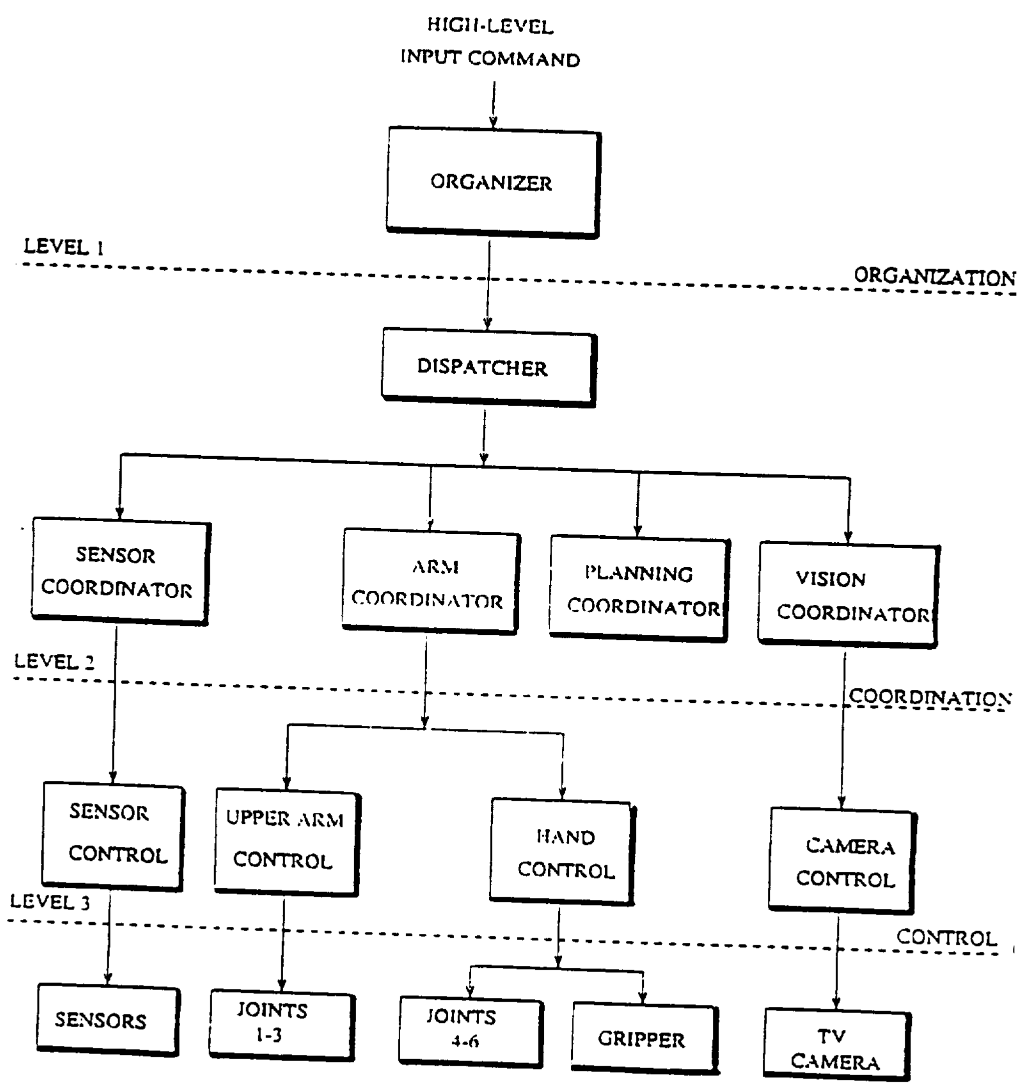
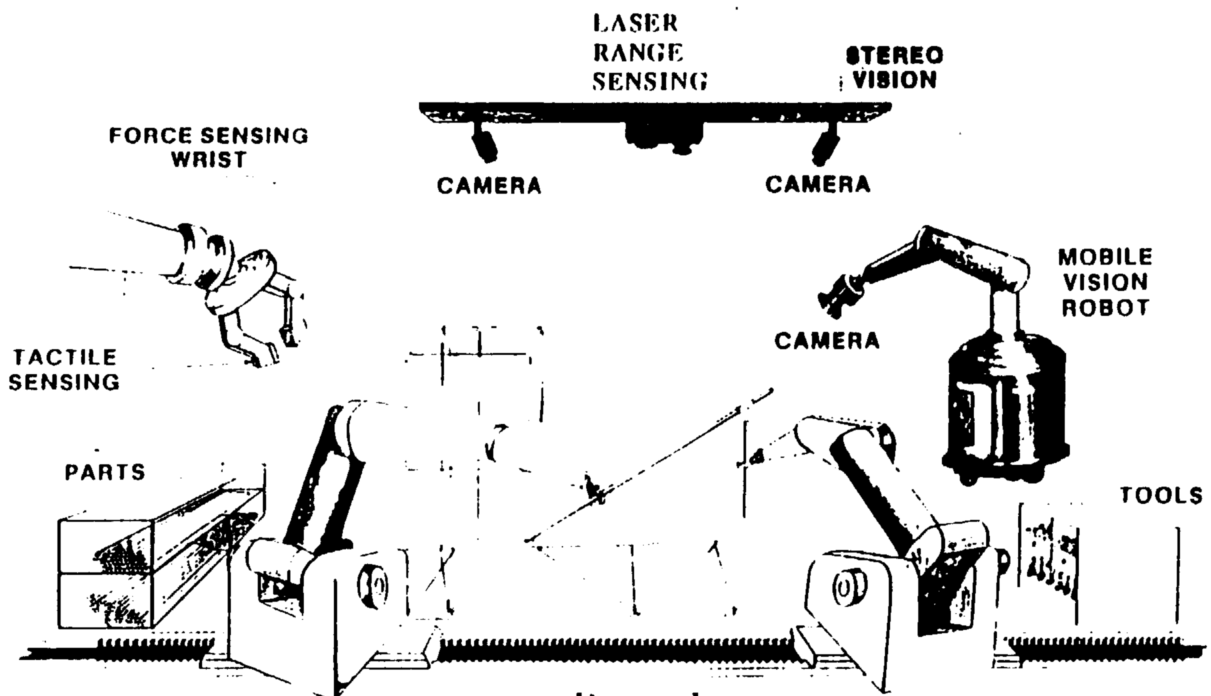


Fig. 9. Hierarchically Intelligent Control for a Manipulator with Sensory Feedback



Rensselaer

CIRSSE LABORATORY

ROBOTIC ASSEMBLY OF IN-SPACE STRUCTURES

FIGURE 10